Real-Time Assessment of Robot Performance during Remote Exploration Operations

Debra Schreckenghost[1], Terrence Fong[2], Tod Milam[1], Estrellina Pacis[3], and Hans Utz[4]
[1]TRACLabs, 1012 Hercules, Houston, TX 77058
ghost@ieee.org, tmilam@traclabs.com
[2] NASA Ames Research Center, Moffett Field, CA 94035
terry.fong@nasa.gov
[3] Space and Naval Warfare Systems Command, San Diego, CA 92152
estrellina.b.pacis@nasa.gov
[4] Research Institute for Advanced Computer Science, Moffett Field, CA 94035
hans.utz@nasa.gov

Abstract—To ensure that robots are used effectively for exploration missions, it is important to assess their performance during operations. We are investigating the definition and computation of performance metrics for assessing remote robotic operations in real-time. Our approach is to monitor data streams from robots, compute performance metrics, and provide Web-based displays of these metrics for assessing robot performance during operations. We evaluated our approach for measuring robot performance with the K10 rovers from NASA Ames Research Center during a field test at Moses Lake Sand Dunes (WA) in June 2008. In this paper we present the results of evaluating our software for robot performance and discuss our conclusions from this evaluation for future robot operations. [1][2]

TABLE OF CONTENTS

1. INTRODUCTION

Future exploration of the Moon will utilize robots for site survey and reconnaissance as well as a variety of lunar surface utility work [6]. Effective use of robots for these applications requires new types of remote operations. For Lunar operations, Earth-based operators will remotely supervise multiple robots performing tasks, independently and jointly. Ground control teams (including scientists and engineers) will monitor the results of these tasks to adjust robot plans. To ensure that robots are used effectively for exploration missions, it is important to assess the performance of these operational models.

We are investigating the definition and computation of performance metrics for assessing remote robotic operations in *real-time*. Our approach is to monitor data streams from robots, compute performance metrics, and provide Web-based displays of these metrics for assessing robot performance during operations. We have identified task performance, reliability, and efficiency metrics for remote robotic operations. We have developed software that performs inline computation of these metrics by monitoring robotic data streams. Metrics are distributed in real-time via a Web server. The current value of metrics can be viewed on dashboard displays. Plots of historical values of metrics overlaid with significant operational events can be viewed on timelines.

We are assessing the usefulness of the performance monitoring software for mission operations. We are working with the Exploration Technology Development Program (ETDP) Human-Robot System (HRS) operations assessment team to evaluate the use of performance metrics during remote science operations. We believe our metrics for robot task performance can be useful to operations personnel in the following ways:

(1) **Mission Manager (MM)** – the person responsible for the success of the mission. We expect metrics summarizing the completion of science objectives to be useful in assessing progress on the mission. We also expect metrics summarizing rover health and productivity to be useful in assessing the effectiveness of rover operations.

(2) **Rover Operator (RO)** – the person who remotely supervises autonomous rover operations and who teleoperates the rover when needed. The Rover Operator will have good insight into ongoing rover operations by virtue of direct participation in those operations. We expect, however, the accumulated metrics (such as Drive Time and Run Time) to provide useful summaries of progress on tasks and rover health

Report Documentation Page

1. REPORT DATE **MAR 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Real-Time Assessment of Robot Performance during Remote Exploration Operations	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Space and Naval Warfare Systems Command,San Diego,CA,92152**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
2009 IEEE Aerospace Conference, 7-14 Mar, 2009, Big Sky, MT. U.S. Government or Federal Rights License

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**12**	

over longer time periods (such as a duty cycle or a mission). We also expect that over time the Rover Operator will gain insight into "typical" rover performance and be able to use metrics as an indicator of degrading or abnormal performance.

(3) **Science Analyst (SA)** – the personnel who define the science plan and analyze the data collected as a result of executing that plan. We expect metrics summarizing progress on data collection by the rover to be useful in helping the science team maintain situation awareness remotely. High-level indicators of rover health also can be informative when science progress is not as expected.

We expect that metrics summarizing the quality of remote communication will be useful to all users, since they indicate the reliability of performance metrics.

We evaluated our approach for measuring robot performance during a field test at Moses Lake Sand Dunes (WA) in June 2008 [6]. We monitored science operations performed at Moses Lake Sand Dunes by two NASA Ames K10 robots from a simulated Mission Control located at JSC and computed task performance metrics for both K10 "Black" and K10 "Red" when performing site survey and reconnaissance tasks. These metrics were available on operational displays for use during the simulated mission. After the test, we used the complete set of data recorded from the robots to compute metrics for robot efficiency and reliability throughout the simulated mission.

In this paper we present the results of evaluating our software for robot performance during simulated science operations at Moses Lake Sand Dunes, and discuss our conclusions from this evaluation for future robot operations.

2. RELATED WORK

The real-time assessment of performance metrics is central to our approach. This requires incremental computation of metrics while operations are ongoing. Other work on the incremental computation of metrics includes the estimation of incremental task progress as a function of time to compute the expected time interval that a robot can function without human attention, called *neglect time* [2].

Real-time assessment of performance also requires strategies to monitor operational context to determine when to compute performance (e.g., infer activity intent). Computation and interpretation of such metrics requires considering environmental, operational, and technology limitations to establish a realistic baseline of comparison. An important conclusion of our research is the identification and categorization of constraints affecting performance baselines (described in Section 6). We have seen examples of such constraints in other research, such as constraints on measuring traverse performance to a single regional terrain type (an example of an environmental constraint) proposed by Tunstel [14]. We know of no work, however, that characterizes these different types of constraints.

The metrics computed during the test at Moses Lake Sand Dunes were informed by prior work on human-robot interaction metrics [1, 5, 11, 15]. We use traversal metrics such as drive time and distance traveled since both site survey and reconnaissance tasks require the robot to traverse a series of waypoints. Our metrics on sensed data differ from the perception metrics discussed by Fong [5], however. The instrument metrics (Lidar and GPR) computed for K10 characterize the collection of data for science return without interpretation of this data. The perception metrics discussed by Fong [5] are intended to characterize the robot's understanding of the environment and thus describe a more active interpretation of sensed data by the robot than the instrument metrics. The real-time computation of performance metrics can be viewed as a form of robot self-awareness, specifically addressing the robot's "capacity for self-monitoring (health, state, task progress)". We have plans to evaluate the computation of real-time metrics onboard the robot in future tests. We have implemented but not yet evaluated measures of human intervention, specifically "Mean Time To Intervene" and the "Mean Time Between Interventions" [16]. We also have a metric for robot productivity derived from metrics proposed for astronaut productivity, the "Work Efficiency Index" (WEI) [7]. WEI is the ratio of robot productive time (i.e., time spent on successful task) and overhead time.

Much of the research on metrics for robotic performance has been for the purpose of assessing and comparing robot technologies [1, 4, 14]. Tunstel [14] defines technology metrics for the Mars Exploration Rovers. His objective is to define metrics characterizing the effects of technology on science return as a baseline for evaluating future rover technologies and establishing baselines for predicting technology impacts on science return. We define multiple different perspectives on metrics that address not only science return (in terms of successful performance of science plans) but also rover productivity and health.

Dillman [4] describes benchmarks for the comparison of robotic technologies. Such benchmarks provide a common basis for understanding technology differences and improvements, and require the execution of a set of common tasks with well-understood environmental state transitions. Similarly benchmarking for diagnostic and prognostic technologies utilizes standardized specifications for fault cataloging and test scenarios [9, 13]. As a result, the baseline for comparison in benchmarking is well understood a priori (i.e., ground truth is known). We are investigating the use of performance metrics for real-time assessment of human-robot operations. These differ from benchmark evaluations (1) by being performed on a broader range of tasks than benchmark tasks, (2) by having less

certain knowledge of the state of the environment and consequent ground truth, and (3) by computing metrics in real-time. Work on benchmarking can however inform what metrics we compute and how we interpret these computations. Technology benchmarks correspond to engineering constraints on performance baselines. Research benchmarks can be relevant to such engineering assessment, but are less relevant to operational and safety assessments.

Metrics for diagnostic and prognostic technologies characterize the ability of the technology to detect or predict problems [9, 13]. They include measures such as detecting a non-existent problem (false positive) or failing to detect a problem (false negative), metrics that consider signal quality [3] and measures of algorithm accuracy [12]. These metrics could be useful in assessing the robot's self-awareness, such as its ability to detect and react to component problems. They also address the assessment of instrumentation accuracy and reliability that could be useful in determining whether to use data measurements when computing our performance metrics. Finally these metrics could be used to assess the ability of our performance monitoring software to correctly detect events triggering the computation of metrics.

Performance monitoring to improve system performance has been utilized in a number of applications other than robotics, including nuclear power [18], process control [17], and traffic flow [19]. Nuclear power plants monitor data to detect faulty or degraded instruments (i.e., instrument performance) and to improve plant performance (e.g., increase thermal efficiency). Such degraded instrument performance can impact plant performance by triggering incorrect operational changes [18]. Process control engineers need metrics that relate controller performance to business objectives [17]. Desborough and Miller categorize performance metrics into two types - business metrics and operational metrics. They identify a requirement for metrics that aid operations in determining when system-wide performance has changed (termed *orientation*). Robotic operations have similar needs for metrics that relate robot task performance (operational metrics) to mission objectives (business metrics). The work described in this paper is a first step toward defining and deploying such metrics for robot operations.

Performance monitoring for space robotic systems does pose challenges not encountered in nuclear power or chemical processing plants. Mobility results in a high degree of interactivity with novel environments. And there is limited ability to alter or constrain the environment to mitigate adverse effects. There are also special challenges in space such as reduced gravity and communication bandwidth and latency. These challenges reinforce the need for techniques such as performance monitoring to aid operations.

3. PERFORMANCE MONITORING SOFTWARE

The purpose of our performance monitoring software is to provide a configurable, reusable system for monitoring robot telemetry, computing metrics about the robot's performance, and presenting these metrics to a user. We support multiple ways of presenting information including: (1) displays of current values of performance metrics, (2) reports that summarize performance over a period of time, and (3) notification about significant performance changes or events. For the recent robotic field test at Moses Lake Sand Dunes, WA, our objective was to evaluate the feasibility of computing and displaying robot performance in real-time. To perform this evaluation, we computed performance metrics for rovers performing remote reconnaissance and site survey activities.

The performance monitoring software evaluated at Moses Lake consists of a computation engine that monitors rover data and computes performance values, data servers for performance values, and Web-based displays. The computation engine passes incoming rover data to the performance algorithms associated with that data. It also can pass the output of one algorithm as input to another algorithm. The computed performance metrics are provided to users via two Apache Tomcat data servers: (1) a *real-time server* that serves the current values of performance computations to dashboard displays developed in the Google Web Toolkit [8] and (2) a *report server* that serves archived values of computations to a timeline display developed using MIT's Simile TimePlot [10]. The dashboard displays provide the user with the most recent values of performance metrics (Figure 1). These displays update automatically when a new value is computed. Values are organized into tables of related metrics, such as "Robot Task Performance" or "Lidar Instrument Performance".
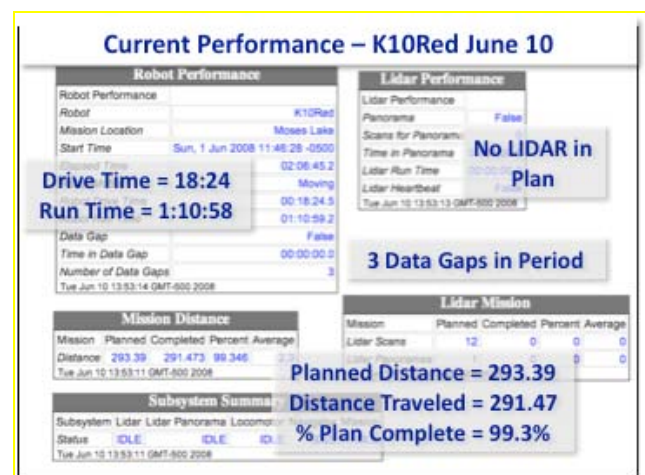

Figure 1 Dashboard Display from Moses Lake Field Test

The timeline display provides the user with a time-based plot of the history of performance values (Figure 2). These

displays update to reflect new values at the user's request. The user can request a current report at any time and can save the report for reviewing later. The user also can change which data should be plotted and which events should be shown in a report. The available choices can be changed using a configuration file.
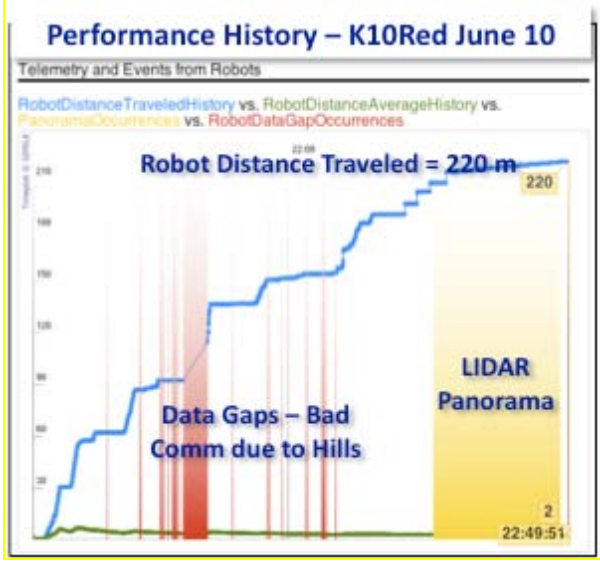

Figure 2. Timeline Display from Moses Lake Field Test

We have developed a library of Java objects that encode robot performance algorithms. XML configuration files specify which algorithms begin executing at system startup. These files identify which algorithms should be actively monitoring data for a particular application and what robot data and computation output are associated with each algorithm. These files also define the value of constants used in the algorithm (such as thresholds). The computation is performed whenever an updated input value is received (either robot data or updated output from another algorithm) and trigger conditions are met. For example, to compute the total distance traveled by a robot, robot pose messages are passed to an algorithm for computing the distance between two poses (TravelBetweenPoints). If the computed distance exceeds a threshold intended to exclude noisy data (.025 meters), the distance value is passed to a second algorithm that keeps a running sum of these distances (DistanceSummation). The resulting sum can be viewed on a dashboard display. A portion of the configuration file for this example is shown below.

```
<Entry>
 <Name>RobotDistanceBetweenPoints</Name>
 <ObjectClass> TravelBetweenPoints</ObjectClass>
 <Config>
  <SubscribeMessage>Pose_Estimate
 </SubscribeMessage>
   <ChangeThreshold>0.025</ChangeThreshold>
 </Config>
</Entry>
<Entry>
 <Name>RobotDistanceTraveled</Name>
 <ObjectClass> DistanceSummation</ObjectClass>
 <Config>
  <SubscribeMessage>RobotDistanceBetweenPoints
```

```
 </SubscribeMessage>
 </Config>
</Entry>
```
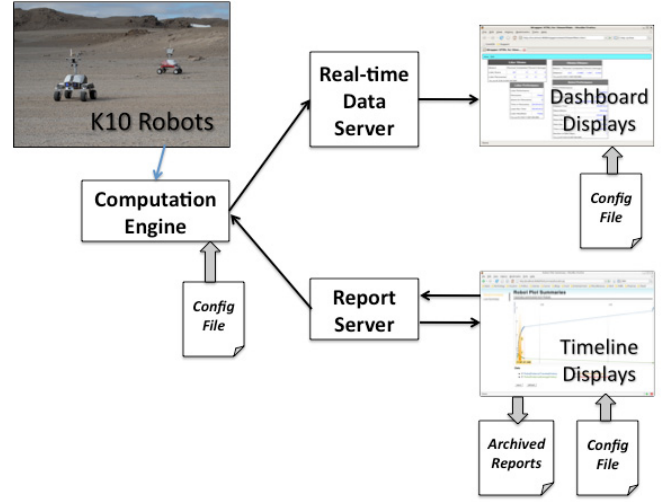
Figure 3 illustrates the architecture of the performance monitoring software.

Figure 3. Architecture for Performance Monitoring

We computed performance metrics for the robot, instruments mounted on the robot, and communication quality during the Moses Lake Sand Dunes field test. These metrics are described below.

Robot Task Performance: measures rover performance on tasks by computing the time spent driving and the distance traveled during that time. Observed performance is compared to estimated performance derived from the rover's plan to indicate whether the robot is performing as expected. See Table 1 for a description of the operational use of these metrics.

$$DriveTime(t) = \sum_{i=1} t_i - t_{i-1} \quad if \ d(x,y) > 0.025m \qquad (1)$$

where
t_i = *time of current pose message*
t_{i-1} = *time of previous pose message if separation > 0.025*
(x, y) = *pose vector*
$$d(x,y) = \sum_{i=1} \sqrt{\left[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2\right]}$$

$$RunTime(t) = \sum_{i=1} t_i - t_{i-1} \quad if \ t_i - t_{i-1} < 10 \sec \qquad (2)$$

where
t_i = *time of current pose message*
t_{i-1} = *time of previous pose message*

$$CompletedDist(x,y) = \sum_{i=1} \sqrt{\left[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2\right]} \quad (3)$$

$$if \ d(x,y) > 0.025m$$

where
(x_i, y_i) = *current robot pose estimate*

$(x_{i-1}, y_{i-1}) = previous\ pose\ estimate$

$$d(x,y) = \sum_{i=1} \sqrt{\left[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2\right]}$$

$$PlannedDist(wx,wy) = \sum_{i=1} \sqrt{\left[(wx_i - wx_{i-1})^2 + (wy_i - y_{i-1})^2\right]} \quad (4)$$

where
$(wx_i, wy_i) = waypoint\ pose\ at\ position\ i$

$$PercentageDistComplete = CompletedDist / PlannedDist \quad (5)$$

$$AvgDistOverTime = CompletedDist / ElapsedTime \quad (6)$$

Table 1. Operational Use of Robot Task Performance

(MM – Mission Manager; RO – Rover Operator; SA – Science Analyst)

Metric	Interpretation	Operational Use
Drive Time	Should be a significant percentage of Run Time for science ops. For site survey, Drive Time may approach Run Time.	**RO**: indicates time rover is on-task
Run Time	Should be significant percentage of the time in duty period	**RO**: indicates if rover is experiencing significant down time
Completed Distance	Should generally increase for science ops	**MM**: indicates task progress **RO**: indicates task progress
Planned Distance	Basis of comparison for Completed Distance	**RO**: helps interpret Completed Distance
Percentage Distance Complete	Should be near 100% for nominal execution of plan	**MM**: indicates progress on science objectives **RO**: indicates how well rover is executing plan **SA**: indicates progress on data collection
Avg Dist / Time	Should be comparable to ops limits on velocity	**RO**: indicates rover mobility performance

Instrument Performance: measures performance of the Lidar instrument during reconnaissance activities. Lidar is used for 3D terrain mapping. During reconnaissance, the rover acquires multiple scans to construct a panorama at specified locations. See Table 2 for a description of the operational use of these metrics.

$$PanoramaIn\Pr ogress(p) = PIP(p) = \begin{cases} true & if\ p = active \\ false & if\ p \neq active \end{cases} \quad (7)$$

where
$p = LidarPanorama\ subsystem\ status$

$$LidarRunTime(t) = t_{end} - t_{start} \quad (8)$$

where
$t_{start} = time\ PIP(p)\ changes\ to\ true$
$t_{end} = time\ PIP(p)\ changes\ to\ false$

$$ObservedPanoramaCount(p) = OPC(p) \quad (9)$$
$$OPC(p) = OPC(p) + 1 \quad if\ p_{i-1} = active \wedge p_i = idle$$
where
$p = LidarPanorama\ subsystem\ status$

$$EstimatedPanoramaCount(k,n) = EPC(k,n) \quad (10)$$
$$EPC(k,n) = EPC(k,n) + 1 \quad if\ k = Lidar\Pr imary \wedge n = true$$
where
$k = name\ of\ task\ in\ Survey\ Manager\ Status\ message$
$n = flag\ indicating\ a\ new\ plan\ has\ been\ uplinked$

$$PercentPanoramaComplete(p,k,n) = OPC(p) / EPC(k,n) \quad (11)$$
where
$p = LidarPanorama\ subsystem\ status$
$k = name\ of\ task\ in\ Survey\ Manager\ Status\ message$
$n = flag\ indicating\ a\ new\ plan\ has\ been\ uplinked$

$$ObservedScanCount(s) = OSC(s) \quad (12)$$
$$OPC(s) = OPC(s) + 1 \quad if\ s_{i-1} \neq active \wedge s_i = active$$
where
$s_i = Lidar\ subsystem\ status\ at\ time\ i$

$$EstimatedScanCount(k,n) = ESC(k,n) = EPC(k,n) * 12 \quad (13)$$
where
$k = name\ of\ task\ in\ Survey\ Manager\ Status\ message$
$n = flag\ indicating\ a\ new\ plan\ has\ been\ uplinked$

$$PercentScanComplete(p,k,n) = OSC(s) / ESC(k,n) \quad (14)$$
where
$s = Lidar\ subsystem\ status$
$k = name\ of\ task\ in\ Survey\ Manager\ Status\ message$
$n = flag\ indicating\ a\ new\ plan\ has\ been\ uplinked$

Table 2. Operational Use of Lidar Performance

(MM – Mission Manager; RO – Rover Operator; SA – Science Analyst)

Metric	Interpretation	Operational Use
Panorama in Progress	Should be true while taking a panorama	**RO**: indicates whether Lidar is functioning normally
Lidar Run Time	Should take ~24 minutes for 12 scan panorama	**RO**: indicates if Lidar performance is typical
Observed Scan Count	Should increase each time a scan completes	**RO**: indicates whether Lidar is functioning normally
Estimated Scan Count	Basis of comparison for Observed Scan Count	**RO**: helps interpret Observed Scan Count
Percentage Scan Complete	Should be near 100% for nominal execution of plan	**RO**: indicates how well rover is executing plan **SA**: indicates progress on data collection
Observed Panorama Count	Should increase when panorama completes	**RO**: indicates whether Lidar is functioning normally
Estimated Panorama Count	Basis of comparison for Observed Panorama Count	**RO**: helps interpret Observed Panorama Count
Percentage Scan Complete	Should be near 100% for nominal execution of plan	**RO**: indicates how well rover is executing plan **SA**: indicates progress on data collection

Communication Quality: measures the quality of remote data communication by detecting when data communication drops out (called *data gaps*)

$$DataGap(tp) = DG(tp) = \begin{cases} t & if \ tp_i - tp_{i-1} > 10 \sec \\ f & if \ tp_i - tp_{i-1} <= 10 \sec \end{cases} \quad (15)$$

where

$tp_i = time \ of \ pose \ message \ i$

$$DataGapCount(tp) = DGC(tp) \quad (16)$$
$$DGC(tp) = DGC(tp) + 1 \quad if \ DG(tp_{i-1}) = false \wedge DG(tp_i) = true$$
where
$tp_i = time \ of \ pose \ message \ i$

Table 3. Operational Use of Communication Quality

(MM – Mission Manager; RO – Rover Operator; SA – Science Analyst)

Metric	Interpretation	Operational Use
Data Gap	Should be true when receiving no data from robot	**All**: indicates if metrics are being updated with robot data
Data Gap Count	Should be relatively low number indicating minimal loss of data	**All**: indicates if quality of metrics might be impacted by loss of data

4. EVALUATION OF ROBOT PERFORMANCE

The performance monitoring software computed performance metrics for the K10 planetary rovers during the Human-Robotic Systems (HRS) field test at Moses Lake Sand Dunes, Washington [6]. The K10 robot was developed by the Intelligent Robotics Group at NASA Ames Research Center. K10 is four-wheel drive and all-wheel steering rover with a passive rocker suspension. The standard sensor suite for a K10 rover includes a Novatel differential GPS system, a Honeywell digital compass, Firewire stereo cameras, a suntracker, and wheel encoders. K10 data interfaces are implemented in CORBA. For the field test at Moses Lake, additional instruments were mounted on both K10 Red and K10 Black to support science operations. K10 Red was equipped for robotic reconnaissance (i.e., using a planetary rover to scout traverses, or sites, prior to EVA activity) with three additional instruments: (1) a 3D Lidar (Optech ILRIS-3D) for terrain mapping, (2) a consumer-grade digital camera mounted on a pan/tilt unit for high-resolution, color panoramas, and (3) a microscopic imager for very high-resolution images of surface materials. K10 Black was equipped with a GSSI SIR-3000 Ground Penetrating Radar (GPR) for 3D subsurface mapping and a microscopic imager for high-resolution surface images to support systematic site surveys.

Planetary science operations were simulated between June 9 – 12, 2008, from an analog ground control at the Johnson Space Center. During that time period, the performance monitoring software was connected to a real-time data stream from the K10 robots while operating at Moses Lake. Rover performance was computed throughout the day, resulting in a daily summary of performance. K10 Red performance was computed for total of 28 hrs and K10 Black performance was computed for a total of 9.5 hrs.

K10 Red Performance

K10 Red performed reconnaissance activities in support of simulated science operations daily from June 9 – 11. A typical reconnaissance performed by K10 Red consisted of navigating to a sequence of waypoints, taking panoramic and microscopic images at each waypoint, and additionally taking Lidar scans at a subset of these waypoints. The drive time for reconnaissance is expected to be less than the run time, since the rover stops while taking instrument readings. For Lidar, taking a full panorama takes around 24 minutes, which can be a significant percentage of the operating time. During the 3 days of reconnaissance operations at Moses Lake Sand Dunes, the K10 Red rover was powered up for 9.35 hrs (Run Time) and drove for 4.2 hrs (Drive Time), or 44.8% of the time it was powered up (Figure 4). The rover attempted to execute eight science plans over the three days and completed four of these plans without human intervention. The total expected distance for these plans was 2235 meters (Planned Distance). The robot actually traversed a total of 2375 meters (Completed Distance) to perform these reconnaissance activities (Figure 5).

Performance metrics also were computed for the Lidar instrument mounted on K10 Red. A typical Lidar activity was taking a panorama. A panorama consists of 12 Lidar scans covering 360 degrees, taken by rotating the robot in 30-degree increments and scanning at each increment. K10 Red took four complete Lidar panoramas during reconnaissance operations, plus two additional scans for a total of 50 Lidar scans. The total time spent taking Lidar during reconnaissance activities was 1.64 hours.
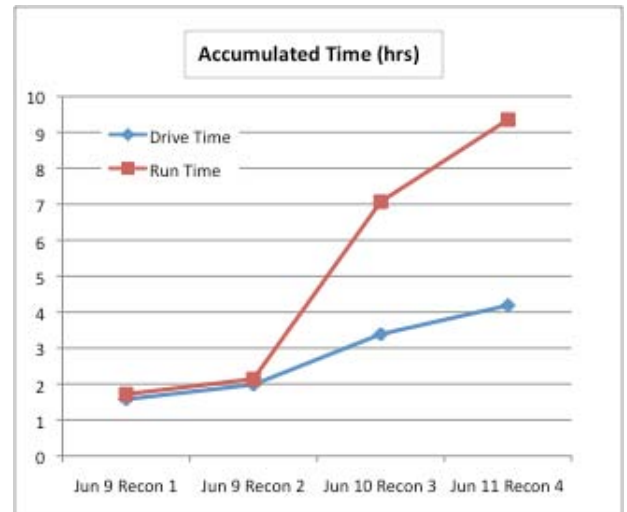

Figure 4. K10 Red Operating Time at Moses Lake

Figure 4 compares drive time (blue line) to run time (red line) for K10 Red. On June 9, drive time and run time are nearly equal, indicating that the robot was driving during most of the duty period as expected. On June 10 and 11, however, the drive time is significantly less than the run time, indicating that the robot was stationary roughly half the time it was running. This mismatch can indicate performance problems, either due to increased robot down time or robot wait time between plans. An inspection of events during these days indicates the robot did experience problems on both days. On June 10, communication difficulties and low lighting resulted in reduced rover performance. The wait time between plans was between 30 and 50 minutes, which is much longer than on June 9. On June 11 the rover base controller had problems and the site terrain impacted traversability, both of which impacted the rover's ability to complete plans in a timely manner.

Figure 5 compares distance traveled by the K10 Red rover to the planned distance. An inspection of this figure indicates a limitation in using accumulated total distance as a metric. By the end of the test, K10 Red had traveled 2375 meters, which agrees well with the planned distance of 2235 meters. A comparison of planned distance to distance traveled for each duty cycle, however, reveals that the rover traveled further than planned on June 9, which masked the fact that the rover traveled less than planned on June 10.
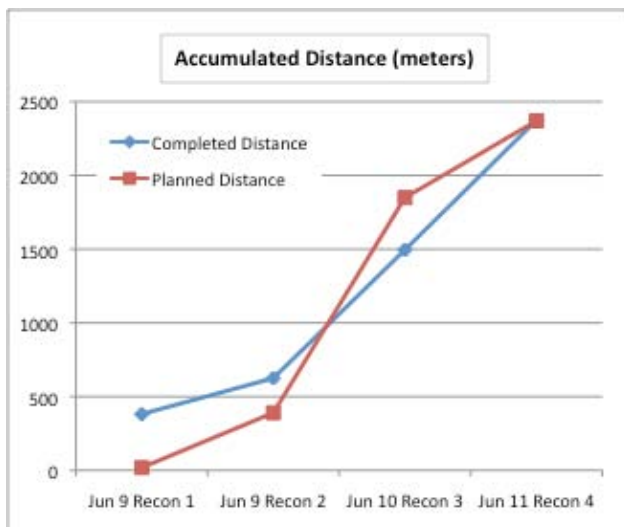

Figure 5. K10 Red Distance at Moses Lake

There were significant problems with the quality of data communication to JSC for remote science operations on June 10 and 11, indicated by the high data gap count on these days. On June 10, there were 54 gaps ranging from 10 seconds to 53.5 minutes and on June 11 there were 65 gaps ranging form 10 seconds to 54 minutes. The reduced data quality affected the accuracy of the performance metrics computed remotely for K10 Red on these days.

K10 Black Performance

K10 Black performed systematic site surveys in support of simulated science operations on June 12. A typical site survey performed by K10 Black consisted of navigating through a sequence of waypoints while taking GPR readings continuously. Often microscopic images were taken at waypoints as well. During the day of survey operations at Moses Lake Sand Dunes, the K10 Black robot was powered up for 5.67 hrs (Run Time) and drove for .64 hrs (Drive Time), or 11.3% of the time it was powered up (Figure 6). Drive Time was lower than expected due a combination of GPR hardware problems and navigation problems due to inaccurate odometry in the sand at the survey site. K10 Black attempted to execute four science plans for a total planned distance of 776 meters (Planned Distance). The robot actually traveled a total of 838 meters (Completed Distance) to perform these survey activities (Figure 7).
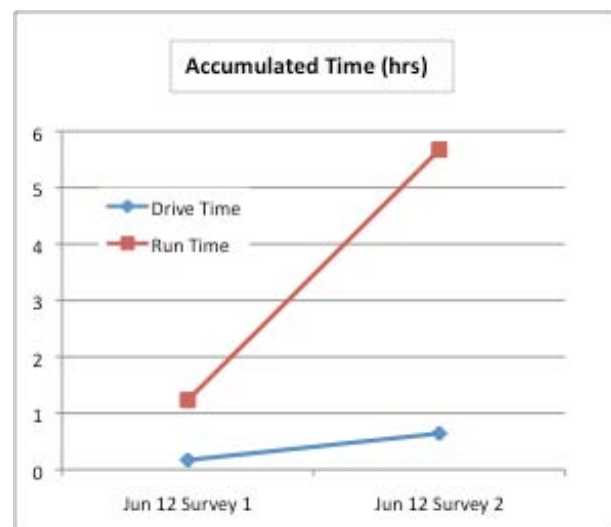

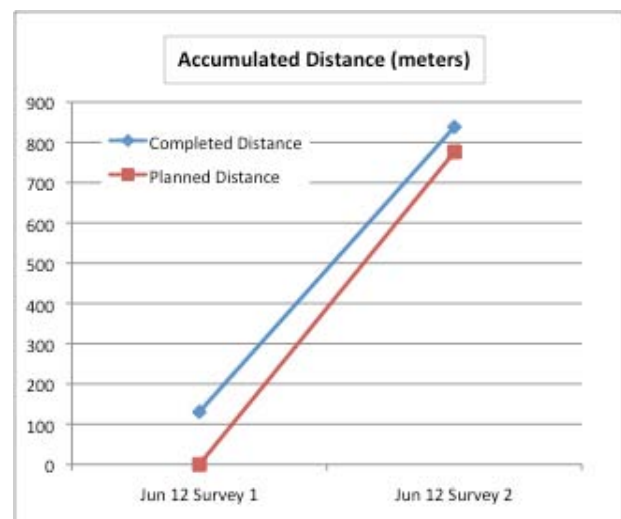**Figure 6. K10 Black Operating Time at Moses Lake**

Figure 7. K10 Black Distance at Moses Lake

The quality of data communication to JSC for remote science operations on June 12 was much better than that on the two days prior. There were a total of 5 gaps, ranging from 10 seconds to 30.75 minutes. These data gaps did not have a significant effect on the accuracy of the performance metrics computed remotely for K10 Black.

5. DISCUSSION

The Role of Context

Understanding the intent of rover activities provides an important basis for grounding the computation and interpretation of performance metrics. In particular, we assert that metrics, whether used for real-time monitoring or post-processed analysis, can only be interpreted in *context*, i.e., with respect to expectations of performance for a particular task, activity plan, robot mode of operation, etc.

For example, the computation of Percentage Distance Complete (ratio of Completed Distance to Planned Distance) requires constraining the computation of Completed Distance to be performed only while K10's plan is executing and to be reset to zero when a new plan is uplinked. Similarly the computation of metrics during autonomous operations should be distinguishable from metrics during teleoperation because expected performance can be quite different in these different control modes. Finally, the computation of metrics during anomalous operations should be separable from nominal operations. The rover may be successful at recovering from a problem while doing less well at achieving nominal mission objectives. In some cases, a metric needs to be computed concurrently in more than one *context*. For example, it is useful to accumulate Completed Distance for each plan (distance traveled within a plan) as well as for the entire mission (total distance traveled).

Another example is that the accuracy of real-time performance computations depends upon the quality of the communication with the robot. We used the Data Gap Count as an indicator of communication quality during the field test at Moses Lake. While useful, this count does not represent information about the duration of communication problems. Based on our experience at Moses Lake Sand Dunes, we have identified a new metric for communication quality. We propose to compute the percentage of the duty period without data from the robot (called Percentage Time in Data Gap) as a measure of the quality of metrics, since a high percentage would indicate lossy communication and reduced confidence in the computed values of performance metrics.

Real-time Displays

During the test we identified and implemented a number of new features for the dashboard displays in response to operational needs. These feature include (1) a user-initiated snapshot of all computed values; we later expect to add the ability to restore the state of the computation engine to this checkpoint state, (2) logging of time-stamped user comments (called *user events*); we later expect to display user events in the timeline display, and (3) user changes to the value of constants in the dashboard displays without restarting the computations. We also added new dashboard displays of rover data during the test. These displays provide information about the rover that aids in understanding performance (i.e., execution status of planned tasks, robot subsystem status). The ability to easily add these new displays indicates the flexibility and extensibility of our architecture for performance monitoring.

Duty Periods

During the HRS field test at Moses Lake Sand Dunes, we computed metrics whenever data were being transmitted from the rover. We did this to ensure that metrics reflected all rover activities. We expected that the value of metrics at the end of each day would then provide a summary of the day's activities. This resulted, however, in the computation of metrics during time periods when the rover was not intended to be operating (e.g., lunch periods, waiting for a demo, etc). As a result, time-based metrics such as Run Time and Drive Time were computed at times when the rover was not intended to be powered up or moving. This resulted in inflated Run Times (see Figures 4 and 7) and artificially reduced the ratio of Drive Time to Run Time.

We believe a more realistic assessment of performance should constrain the computation of metrics to time periods when the rover is intended to be performing tasks. We define a concept called a *duty period* that corresponds to a contiguous interval of operations (such as from rover startup in the morning until a planned ground control break period). We propose to compute accumulated metrics only during duty periods in future tests.

Technology Deployment at NASA

The evaluation of our performance monitoring software during robotic field tests has provided useful feedback on the utility of the proposed metrics. In particular we have gained valuable experience in using such metrics when data are subject to periodic dropouts. We recently evaluated the use of our performance metrics during an analog test of rover reconnaissance as a precursor to planning astronaut EVAs where we got feedback about the utility of metrics for flight operations. There is remaining work, however, before this technology can be operationally deployed.

A variety of data types are used by the real-time performance monitoring of robots. These data types have correlates in current NASA operations, including spacecraft telemetry, Comps (combinations of raw telemetry), planning information, and flight rules. Deployment of performance

monitoring technology in flight operations will require that such information be captured electronically and made available for computer-based reasoning. Recent efforts to move flight products like Space Station activity plans and attitude timelines into XML supports this objective and future programs are investigating standards for representing such information, but current NASA programs have not yet achieved the level of information integration that will be needed to automate performance monitoring as described in this paper.

The availability of real-time performance data raises the possibility of using this data to adjust operations for improved performance. Such real-time performance management has become a consideration for commercial plant operations as plant automation is introduced. Often the barriers to real-time performance management are as much cultural as technical. In a recent article (Spiegel, 2007), it was observed the adjustment of plant operations by the enterprise has met strong resistance due to mutual mistrust between plant operators and corporate Information Technology departments. Such barriers are also a possibility when introducing real-time performance management into NASA operations.

6. FUTURE WORK

Assessing Performance Based on Expectations

An important result from this test is an approach for improving the interpretation of performance metrics. Prior to the test, we established performance expectations from robot task plans (called plan performance). Satisfactory performance was defined as the achievement of planned targets (e.g., successful completion of planned tasks or collection of planned samples). However, observation of robot performance during the test made it clear there are other considerations in defining satisfactory performance.

The limits imposed by robotic hardware and software design define expectations about the robot's engineered performance. Engineered performance defines robot capabilities under ideal circumstances. Robot operations, however, often occur under less than ideal circumstances, which can constrain maximum speed and sensor accuracy. The limits imposed by the environment in which the robot operates define expectations about robot operational performance. Engineered performance also can be degraded over the course of multiple missions due to normal component wear as well as systemic problems that reduce performance.

The limits imposed by component use and subsystem problems define the expectations about robot degraded mode performance. Robot operations can be further constrained by flight rules. Maximum robot speeds may be reduced when operating near other robots. Or, certain robot behaviors may be precluded when operating near humans. The limits imposed by the flight rules established for a mission define expectations about safe performance.

Moreover, when comparing robot metrics over multiple duties periods or missions (called historical performance) or comparing the performance of different robots, it is important to establish a basis of comparison that identifies which of these performance dimensions predominate and how these dimensions combine to establish performance expectations, such as: Did robotic hardware or software change between missions? And were the robots being compared operating under the similar operational constraints?

Summarizing Performance

We are currently preparing to use our real-time performance monitoring software to support a test of science operations at NASA Ames Research Center in November 2008. Based on our experience at Moses Lake Sand Dunes, we have identified a few key metrics that should provide a high-level summary of rover performance. These metrics will be displayed on a summary Web page for operational use. We will summarize rover performance from the following three perspectives:

- Mission: metrics that describe the rover's contribution to the mission. The Work Efficiency Index [7] shows the ratio of rover productive time to rover overhead time. When WEI exceeds 1.0, the rover is spending more time accomplishing mission objectives than performing other activities. The Percentage of Time on Task shows the ratio of rover productive time to total operating time.

- Science: metrics that describe the rover's performance with respect to the science plan. The Percentage Distance Complete summarizes the percentage of the planned distance that has been traveled by the rover. The Percentage of Tasks Complete describes the percentage of planned tasks successfully completed.

- Rover: metrics that describe the rover's health and status. We detect when personnel intervene in rover operations to handle anomalies, and we compute the Mean Time to Intervene and the Mean Time between Interventions.

We also plan to compute the Percentage Time in Data Gap as a measure of the quality of computed values, since a high percentage indicates lossy communication and a lower confidence in metrics.

To illustrate how we expect this information to summarize performance for mission managers, consider the following example based on data collected by K10 Red at Moses Lake on the afternoon of June 11, 2008. In this example, we

played back recorded data into our performance monitoring software and took a screen shot of the Ops Synopsis dashboard display partway through the run (Figure 8). In this situation, the rover is executing a science plan to perform reconnaissance of the Moses Lake Sand Dunes. At the time of the screen shot, the rover has spent 59% of its time on planned tasks for a WEI of 1.446. This indicates the mission is going reasonably well, since WEI is greater than 1.0 and the rover is spending the majority of its time on planned tasks. The science metrics indicate good progress is being made on data collection, requiring the rover to drive 326 meters to complete 70% of the plan. The rover metrics indicate minimal unintended human intervention, with MTBI (00:39:10.7) significantly greater than MTTI (00:02:01.2). The Percentage Time in Data Gap is zero, indicating that metrics are based on all available robot data and are thus considered reliable.

Ops Synopsis	
Ops Synopsis	
Start Time	Wed, 11 Jun 2008 17:54:50 -0500
Event Name	Moses Lake Field Test
Objective	Recon
PET	01:22:23.9
Location	Sand Dunes
Plan	MLT
Robot	K10 Red
WEI	1.446
% Time on Task	59.122
Distance Traveled	326.294
% Plan Complete	70.27
MTTI	00:02:01.2
MTBI	00:39:10.7
Data Dropout	False
% Time in Dropout	0
Wed Oct 29 15:53:51 GMT-500 2008	

Figure 8. Example of Summary Metrics for Mission Manager

7. CONCLUSIONS

Our performance computation software operated reliably for over 46 hours during a four-day period. Task performance metrics computed for the K10 robots during the field test include robot drive time, robot run time, and performance during plans (e.g., percentage of planned distance completed, percentage of planned samples complete). We measured the number of times we lost signal from the robot at the simulated Mission Control as a measure of the quality of our real-time metrics. Based on these results, we conclude that our proposed approach for computing and displaying rover performance metrics in real-time is viable.

8. ACKNOWLEDGEMENTS

REFERENCES

[1] Nicola Basilico and Francesco Amigoni, "On Evaluating Performance of Exploration Strategies for Autonomous Mobile Robots," Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems Proceedings, IEEE International Conference on Intelligent Robots and Systems, September 26, 2008.

[2] Jacob Crandall, Michael Goodrich, Dan Olsen, Jr., and Curtis W. Nielsen, "Validating Human–Robot Interaction Schemes in Multitasking Environments", IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 35, No. 4, July 2005.

[3] H. Depold, J. Siegel, and J. Hull, "Metrics for Evaluating the Accuracy of Diagnostic Fault Detection Systems", IGTI Turbo Expo, June 2004.

[4] Rudiger Dillmann, "KA 1.10 Benchmarks for Robotics Research," European Robotics Network (EURON), IST-2000-26048, April 24 2004.

[5] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schulz, and M. Goodrich, "Common Metrics for Human-Robot Interaction," 2006 ACM Conference on Human-Robot Interaction. March 2004.

[6] T. Fong, M. Bualat, M. Deans, M. Allan, X. Bouyssounouse, M. Broxton, L. Edwards, R. Elphic, L. Fluckiger, J. Frank, L. Keely, L. Kobayashi, P. Lee, S. Y. Lee, D. Lees, E. Pacis, E. Park, L. Pedersen, D. Schreckenghost, T. Smith, V. To, and H. Utz, "Field Testing of Utility Robots for Lunar Surface Operations, " AIAA Space 2008, AIAA-2008-7886. September 2008.

[7] Mike Gernhardt, Work Efficiency Indices. Presentation at Johnson Space Center. November 15, 2005.

[8] Google Web Toolkit http://code.google.com/webtoolkit/).

[9] Tolga Kurtoglu, Ole J. Mengshoel, and Scott Poll, A Framework for Systematic Benchmarking of Monitoring and Diagnostic Systems," 2008 International Conference on Prognostics and Health Management Proceedings, October 6-9, 2008.

[10] MIT Simile http://code.google.com/p/simile-widgets/

[11] D. Olsen and M. Goodrich, "Metrics for Evaluating Human-Robot Interactions," Proceedings of NIST Performance Metrics for Intelligent Systems, Sep 2003.

[12] SAE (Society of Automotive Engineers) E-32, "Health and Usage Monitoring Metrics, Monitoring the Monitor", SAE ARP 5783, February 2008.

[13] Abhinav Saxena, Jose Celaya, Edward Balaban, Kai Goebel, Bhaskar Saha, Sankalita Saha, and Mark Schwabacher, "Metrics for Evaluating Performance of Prognostic Techniques," 2008 International Conference on Prognostics and Health Management Proceedings, October 6-9, 2008.

[14] Edward Tunstel, "Performance Metrics for Operational Mars Exploration Rover," Journal of Field Robotics, Volume 24 Issue 8-9, 651 – 670, September 2007.

[15] H. A. Yanco, J. L., Drury, and J. Scholtz, "Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition," Journal of Human-Computer Interaction, Volume 19, 117—149, 2004.

[16] J. Arnold, "Towards a Framework for Architecting Heterogeneous Teams of Humans and Robots for Space Exploration", M.S. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology. 2006.

[17] Lane Desborough and Randy Miller, "Increasing Customer Value of Industrial Control Performance Monitoring -- Honeywell's Experience," Sixth International Conference on Chemical Process Control, Jan 2001.

[18] J.W. Hines and E. Davis, "Lessons Learned from the U.S. Nuclear Power Plant On-line Monitoring Programs," Progress in Nuclear Energy, Volume 46, Issues 3-4, 2005, pp 176-189.

[19] Henry X. Liu, Wenteng Ma, Xinkai Wu, and Heng Hu, "Development of a Real-Time Arterial Performance Monitoring System Using Traffic Data Available from Existing Signal Systems," Minnesota Department of Transportation Technical Report MN/RC 2009-01, December 2008.

[20] Rob Spiegel, "Connecting Sensor Data to the Boardroom," Automation World, March 2007.

BIOGRAPHY

Debra Schreckenghost is a Senior Scientist with TRACLabs in Houston, TX. She is Principal Investigator of the NASA Phase II SBIR developing a software framework for inline monitoring of robot health and performance. Previously she has led projects developing adjustable autonomy software for monitoring and control of crew life support systems and personal agents for distributed space

operations. She has published over 70 technical publications in the areas of adjustable autonomy, architectures for intelligent software agents, and human-computer interaction. She received a BS in electrical engineering from the University of Houston and an MS in electrical engineering from Rice University.
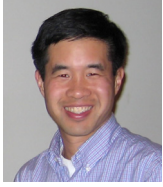
Terrence Fong is the director of the Intelligent Robotics Group at the NASA Ames Research Center. Prior to this, he was the deputy leader of the Virtual Reality and Active Interfaces Group at the Swiss Federal Institute of Technology, Lausanne. Fong has published more than fifty papers in field robotics, human-robot interaction, virtual reality user interfaces, and parallel processing. He received his BS and MS in Aeronautics and Astronautics from the Massachusetts Institute of Technology and his Ph.D. in Robotics from Carnegie Mellon University.

Tod Milam is a Research Scientist with TRACLabs, Houston, TX. Previously, he was a Senior Software Engineer at Lockheed Martin Space Mission Systems & Services, working on traffic management systems, as well as software for Mission Control Center at Johnson Space Center. He began working with TRACLabs in 2000 and was a key developer of TRACLabs' Distributed Collaboration and Interaction System, providing intelligent agents for human-automation interaction in mission operations. He is currently the lead developer for a NASA Phase II SBIR on developing software for in-line computation of robot measures of performance. Mr. Milam received his Bachelors degree in Computer Science from Drake University in 1988..

Estrellina Pacis is a Project Manager at SPAWAR Systems Center, Pacific where she has been developing unmanned systems for the past 6 years. Mrs. Pacis has led numerous projects focused on the development, integration, and testing of autonomous behaviors for military applications. Her current focus is on the study of robotic mapping and exploration techniques suitable for supporting dismount troops during urban operations.

Hans Utz is a Mobile Robot Software Architect for the Research Institute for Advanced Computer Science (RIACS) working with the Intelligent Robotics Group at NASA Ames Research Center (ARC). His work is concerned with the development of advanced software architectures for autonomous mobile robotics. Before joining NASA ARC, he was a PhD student at the Dept of Neuroinformatics, University of Ulm, Germany. He designed the robotics middleware Miro and coached the Ulm Sparrows, a project on "autonomous mobile robotics in highly dynamic environments", which is probably better known as robot soccer. He has a PhD in Computer Science from the University of Ulm, Germany.